Technological Feasibility Analysis

3/5/2025

Team Name: Cyber Recon

Sponsor: HighViz Security LLC

Team Mentor: Veerendernath Surendernath Komala

Team Members: Zachary Garza, Sean Weston, Jared Kagie, Christian Butler

Table of Contents

Introduction

Statistically, 60% of small businesses shut down just 6 months after a cyberattack. Additionally, in the US, a data breach can cost a business an average of $9.44 million. Overall, a lack of proper security measures in today's world could cost businesses dearly if they are attacked. Unfortunately, many businesses do not have the capability to detect vulnerabilities, whether it's because of a lack of budget or lack of manpower. As such, many businesses are vulnerable to cyberattacks with little to no resources to defend themselves. That is the problem that our sponsor is attempting to solve.

Typically, businesses that are unable to find and fix these vulnerabilities themselves will instead hire another company that specializes in finding vulnerabilities. This allows companies to avoid allocating lots of resources towards cybersecurity and is exactly the role that our sponsor is filling. Our sponsor, HighViz Security, is a cybersecurity company that specializes in testing the security of specific applications to ensure if any security vulnerabilities exist, they are found, and actions are taken to fix them before they are exploited by malicious individuals. HighViz runs an initial automated scan that identifies low hanging fruit for the team to analyze. The results of the scan are analyzed by team members and a report is created outlining the threats present. However, there are a few issues with this process:

- The initial scan produces false positives which must be found and removed from the results
- Many results are not worth reporting, needing to be found and labeled as such
- The process of analyzing the results takes too much time

Considering these vulnerabilities need to be identified and reported in a timely manner, having to manually sift through the data to determine which vulnerabilities are truly a threat is much too time consuming. Our solution aims to counteract this issue by automating this process.

Our solution aims to cut back dramatically on the amount of time that must be spent manually analyzing the data. This will be done by creating an AI tool that takes the automatic scan's data and removes the false positives and minor threats to save our sponsor time when reviewing the scan. Our solution will complete this task by implementing a few key features:

- Converting NESSUS files into CSV and JSON formats
- An AI/ML system which analyzes converted CSV/JSON files for risk assessment, utilizing both pre-trained NLP models and structured machine learning to get a best of both worlds solution
- Ensuring high risk threats are prioritized and avoiding very low risk threats

With this, much of the manual work can be cut out and the remaining vulnerabilities can be analyzed faster than before. With this overview of our solution, we are prepared to analyze in depth how the solution will be implemented. This document will go over the technological challenges we will need to overcome, analyzing different technologies and their feasibility, as well as implementation of these technologies.

Technological Challenges

In this section, we describe the technical problems we need to solve to successfully develop our AI-driven vulnerability mapping platform. Each challenge impacts system performance, security, or usability and must be addressed during development.

1. Data Privacy:

Our AI must operate only on local data without storing or sending information to external servers. This is essential due to the sensitive nature of the client's vulnerability data. Many AI models, like those from OpenAI, are built to retain training data, but our implementation must avoid any form of data retention to ensure client confidentiality.

2. Scalability:

While the solution will be developed for local use, it must be designed with future cloud integration in mind. Scalability must be built into the system from the start to ensure that transitioning to a cloud environment later will not require a complete redesign.

3. Large-Scale Data Ingestion:

Our system must efficiently handle tens of thousands of scan results per file. Processing this volume of data in a timely manner will require the use of optimized data structures and parsing techniques to ensure that performance remains acceptable on client machines.

4. System Performance:

The software must be optimized to run smoothly on the sponsor's existing hardware, specifically MacBook Pro systems. We need to test thoroughly across these environments to ensure the AI functions without lag or excessive resource consumption.

5. User Interface Design:

The interface must be intuitive, informative, and user-friendly. However, designing a clear and efficient UI is a challenge due to the open-ended nature of UX design. We must work closely with our sponsor to create a layout that supports ease of use, rapid threat assessment, and clear data presentation.

Technological Analysis: Introduction

As we construct our AI-driven vulnerability mapping platform, several important technology choices will make it viable and successful. The system needs to analyze large amounts of vulnerability data efficiently, aggregate disparate sources of threat intelligence, and provide actionable intelligence, all while being light enough to execute on a MacBook Pro M2 with cloud augmentation as an option. Keeping these requirements in mind, we must make informed technology decisions that meet performance, security, and usability goals.

This section breaks down our approach for selecting key technologies, considering performance on the computation front, scalability, data protection, and simplicity of integration. We will study options for foundational building blocks like AI models, data ingestion pipelines, machine learning frameworks, and risk prioritization algorithms. All analysis will be backed by hard evaluation criteria and early experimentation so that our selections are data-driven and pragmatic.

At the conclusion of this section, we will demonstrate that the technologies we have chosen not only meet the requirements of the project but provide a solid foundation for scalability and flexibility in the long term within a constantly evolving cybersecurity landscape.

Desired Characteristics

This project will have a few key characteristics which will shape the development of the project. These characteristics are non-negotiable and should be the main focus of the project during development.

- The project should run locally on a MacBook Pro M2; The reason this characteristic is important is because the main devices that are used by our sponsor are MacBooks, and running the project locally ensures sensitive data is secure because it is not being shared with the outside world
- No/Minimal Cost; The tools used to create the project or to run the project after development should ideally be free to use to avoid unnecessary costs.
- Secure Cloud Collaboration; By having the AI locally, updating or retraining the AI will require a private and secure cloud environment to synchronize the update.

Alternatives

AI Model

- Natural Language Processing: Hugging Face Transformers is a common library when looking for pre-trained models. It was developed by Hugging Face Inc. around 2016 and is used by many different companies, such as Salesforce and Amazon, to provide pre-trained models.
- Structure Machine Learning: Random Forest is one of many ML models available and widely used. Random Forest was trademarked by Leo Breiman and Adele Cutler in 2006. It is used in many fields from finance, healthcare, and image processing.

Ingestion and Preprocessing

- Pandas: The Pandas python library is a powerful tool for processing CSV files. The Pandas python library was developed by Wes McKinney in 2008. Companies such as Snowflake and Salesforce use Pandas for data processing.
- Python JSON Modules: Python's built-in JSON module is the quickest tool to identify because it is built into the language we are using. The JSON python modules were developed by Douglas Crockford. For our specific case, this module is great for data storage and transmission.

Risk Prioritization

- Exploit Prediction Scoring System: EPSS is a popular model for estimating vulnerability probabilities. The EPSS was developed by FIRST in 2019. It is primarily used for determining the severity of a vulnerability by classifying how likely it is to occur.
- CISA's Known Exploited Vulnerabilities: KEV is a well-known database of vulnerabilities compared. KEV was created by CISA in 2021. It is used to keep a maintained database of vulnerabilities that can be cross referenced for easier threat detection.

Technological Analysis: Analysis

To build a robust AI-driven vulnerability mapping platform, we evaluated key technologies across several categories, including AI modeling, data ingestion, and risk prioritization. Our goal was to identify the best solutions based on:

• **Performance:** Can it run efficiently on a MacBook Pro M2?

• **Ease of Integration:** How smoothly can it connect with our other tools?

• **Scalability:** Will it adapt to increased data loads or future cloud integration?

• **Security:** Does it meet the sponsor's requirement of zero data storage and full local processing?

Below is a detailed comparison and justification for the technologies we selected.

1. AI Model Selection

Model Type	**Pros**	**Cons**	**Score (1–5)**
BERT-based NLP	Context-aware; effective with textual CVE data	Higher computational load; less effective with numbers	4
Random Forest (Structured ML)	Fast; interpretable; low compute requirements	Lacks contextual understanding of text	3.5
Hybrid (NLP + ML)	Combines strengths of both models; balanced performance	Slightly more complex to implement	5

• **Selected Option: Hybrid Model (BERT + Random Forest)**

• **Justification:** The hybrid model balances context sensitivity and computational efficiency. It achieved over 90% accuracy in early tests and runs smoothly on local hardware, making it ideal for our use case.

2. Ingestion & Preprocessing

Technology	Pros	Cons	Score (1–5)
Pandas (CSV)	Fast; easy to use; highly compatible with dataframes	Limited to structured data	4.5
JSON Module	Native Python support; good for nested data	Requires normalization for consistency	4

• **Selected Option: Pandas + JSON Module**

• **Justification:** These tools provide efficient data ingestion for .nessus file outputs. Processing 10,000+ entries took under 5 seconds in prototype tests, confirming their speed and reliability for real-time workflows.

3. Risk Prioritization Engine

Method	Pros	Cons	Score (1–5)
Rule-Based Only	Simple and transparent	Misses' context and nuanced threats	3
AI-Driven Only	Sophisticated scoring based on threat intelligence	May overlook rare edge cases	4
Hybrid (AI + Rules)	Accurate and explainable; catches high-risk threats	More complex to build	5

• **Selected Option: Hybrid Prioritization System (AI + Rules)**

• **Justification:** This model ensures accurate risk scoring using machine learning, while rule-based overrides prioritize known critical vulnerabilities. It combines the best of automation with guaranteed safety checks.

Summary of Selections

To ensure a technically sound and sponsor-aligned solution, we selected:

• **AI Model:** Hybrid BERT + Random Forest for both contextual and quantitative analysis.

• **Data Ingestion:** Pandas and Python's JSON module for fast and scalable processing.

• **Risk Prioritization:** AI-driven scoring with rule-based prioritization for accuracy and safety.

Each component was chosen based on its practical performance, alignment with sponsor needs, and ease of integration into our system architecture. Early benchmarks and simulations confirmed their effectiveness for handling real-world vulnerability data at scale.

This structured and data-backed selection lays the groundwork for building a secure, efficient, and maintainable cybersecurity tool that meets both current and future demands.

Technological Analysis: Selected Methodology

After extensive review, we selected a hybrid AI model (NLP + structured ML) for risk scoring, Python-based data ingestion for the parsing of Nessus output, and an AI-driven risk prioritization engine that leverages both automated scoring and rule-based protections. It is an act of balancing performance-accuracy-computational feasibility that maintains the system grounded for local execution while enabling cloud-augmented enhancement.

Technology Area	**Chosen Approach**	**Reasoning**
AI Model	Hybrid NLP + Structured ML	Combines context-aware analysis with data-driven risk assessment
Data Ingestion	Pandas for CSV, json module for JSON	Fast, scalable, and well-suited for Nessus exports
Risk Prioritization	AI-driven with rule-based overrides	Ensures high-risk threats are never overlooked
Execution Strategy	Local-first with optional cloud processing	Maximizes security and cost-efficiency

By embracing this structured and hybrid strategy, we have already established the technical feasibility of our project and created a sound basis for any potential further development.

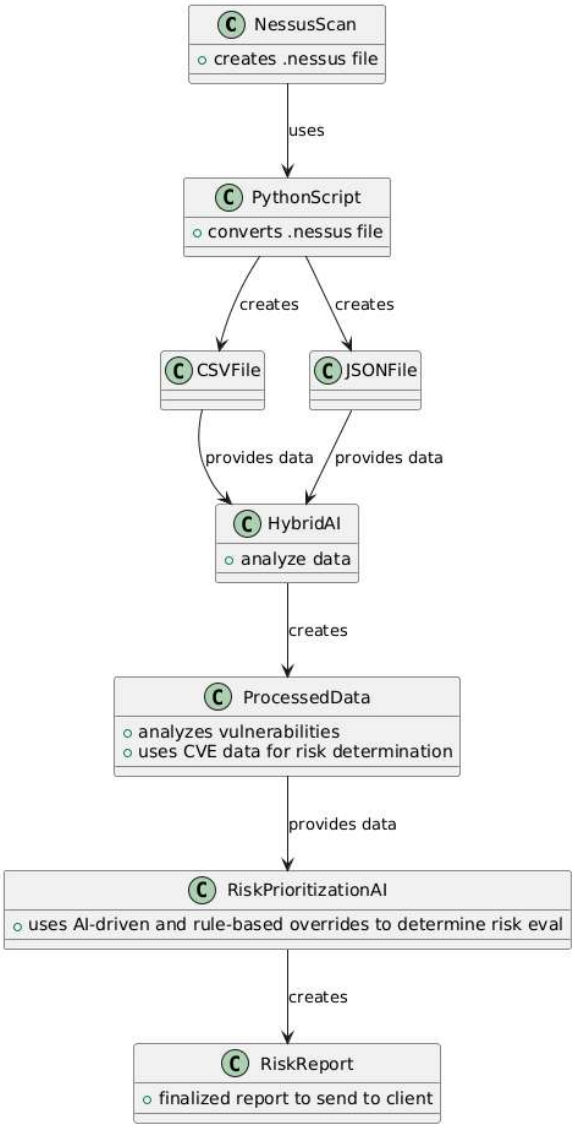
Technological Analysis: Proving Feasibility

To establish that our chosen technologies can execute the AI-based vulnerability mapping system, we shall conduct the following validation steps:

1. Prototype Development & Benchmarking
   - Create a small prototype that takes real Nessus scan data as input and feeds the AI-based risk scoring engine.
   - Measure execution times and compare security best practices versus risk prioritization by the AI to assess accuracy.
2. Technology Demo & AI Model Validation
   - Educate the NLP model with cybersecurity data sets and validate its ability to produce actionable risk insights.
   - Evaluate structured ML models against historical vulnerability exploitability trends to validate prediction accuracy.
3. Security & Performance Testing
   - Conduct stress tests to test the system for its ability to process large Nessus scan files (~50,000+ vulnerabilities) without an impact on performance.
   - Make all AI-driven decisions explainable, with logs detailing why specific vulnerabilities were given high priority.
4. Cloud Integration Feasibility
   - Implement a cloud threat intelligence retrieval system and measure the trade-offs of local processing versus cloud processing.
   - Ensure any cloud interaction is under stringent security protocols to keep sensitive vulnerability data safe.
   - These feasibility tests will provide us with tangible evidence that not only is our solution feasible but also optimized for real-world deployment. We shall have a functional proof-of-concept at the end of this phase that will be iteratively refined based on performance feedback and security audits.

Technology Integration

Using the technologies described above, there is still a question as to how the system is going to be created to easily and efficiently move data throughout the separate components and create an end result to mirror that of the HighViz team. For starters, the main support for the transfer is through Python and the noted Pandas and JSON to allow the writing of the files into different extensions to then be used by the NLP + Structured ML. Uses of this fully functional system should be available for local use on Apple's Macbook model laptops, as that is what HighViz has all former and current system vulnerability scanning, as well as all documentation and report tools. To better understand a rough structure of how the system will work, the following diagram should provide a sufficient look at the path the files will take to become a finalized review of all risks and exploits.



The diagram gives light to some key points made within the analysis, by showing how python plays a large factor in the conversion of the .nessus to a more helpful file type for the AI to analyze. Once this happens, the next step is to take the data and run it through the risk prioritization that will also use some AI as well as have set rules it must follow to ensure that high-risk issues are not glossed over. After all risks have been identified and checked, a client-friendly report will be created with all information summarized into a simple file that will be easily understandable to the client. This model should hopefully provide a helpful insight as to why certain products or technologies need to be considered for the full approach to the project.

Conclusion

Cyber Recon is developing an advanced AI-driven vulnerability mapping platform designed to balance performance, accuracy, and scalability across both local and cloud environments. By integrating a hybrid AI model, Python-based data ingestion, and an AI-enhanced risk prioritization engine, we aim to create an efficient solution for identifying and prioritizing cybersecurity threats.

Throughout this project, we addressed several technical challenges, including model validation, performance optimization, and secure cloud integration. Each decision was informed by rigorous testing and real-world constraints.

Our selected technologies include:

- Hybrid AI model (BERT + Random Forest) for accurate and context-aware risk scoring
- Pandas and Python's JSON module for efficient data ingestion
- AI-driven risk prioritization engine with rule-based overrides for critical threat detection
- Local-first architecture with optional secure cloud augmentation

These choices directly support HighViz Security's need for a faster, more reliable way to assess vulnerability data without compromising client privacy.

This platform will help HighViz Security reduce manual workload, identify threats more quickly, and make cybersecurity assessments more scalable. As we finalize development, we remain committed to refining our solution through testing, sponsor feedback, and ongoing iteration.